PCT

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification [7] : **G06F 1/00**	**A1**	(11) International Publication Number: **WO 00/28398** (43) International Publication Date: 18 May 2000 (18.05.00)

(54) Title: CONTENT SUPPLIED AS SOFTWARE OBJECTS FOR COPYRIGHT PROTECTION
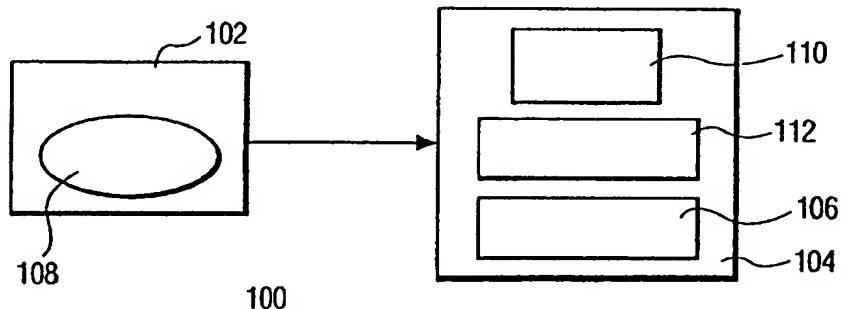
(57) Abstract

Content information such as a movie, an audio file or a textual message is supplied to an end–user in a software object that has an encapsulated procedure for end–user access of the content information in a runtime environment. The object can specify time frame for and manner wherein the content information is to be accessed. Since the procedure is encapsulated in the object together with the content data, and since transport of the object over the Internet is done after serializing, an adequate degree of security is provided against unauthorized play–out or copying.

Content supplied as software objects for copyright protection.

The invention relates to a method of providing controlled access to content data, in particular as provided to an end-user of consumer electronics equipment or a multimedia computer. The invention also relates to a method of protecting unauthorized copying of digital data content.

5   The terms "content" and "content data" as used herein refer to data that contains information for the end-user, e.g., alphanumerical text, graphics, video, audio, multimedia, etc.

Currently, digital content is usually sent or streamed to the end-user from a remote server using cable or Internet, or is provided on a DVD, etc. The content is processed by the receiving device (e.g., a computer, a set-top box) and rendered to the end user. Content

10   can be copyright-protected using "watermarking": adding digital modifications that are undetectable by the user, but are recognized by the rendering or verifying software or hardware. Another way to protect the content is encryption. The content data is encoded using special encryption algorithms before sending the data to its destination. At the destination device or and at an intermediate processor (PC, cable box, set-top box, etc...) the stream is

15   decrypted before rendering or storage.

There are a number of problems associated with these methods. Watermarking, for example, is highly sensitive to content modifications. If the digital content is modified by software/hardware, which is not aware of the "watermark", the watermark is not guaranteed to persist in the original form. For example, picture resizing, storing the picture in a different file

20   format, applying compression techniques, etc., affect also the watermark and make it less useful. Encryption methods also provide a number of challenges. As computer/processing hardware and software become more powerful and sophisticated, well-known algorithms get "cracked" and are rendered obsolete. Another challenge is standardization. That is, in order to address the mass market with multiple vendors, a content protection method needs to be

25   standard. On the other hand, a standard encryption method becomes obsolete relatively fast, is known to specialists and therefore is easy to defeat.

It is an object of the invention to provide an alternative method for providing controlled access to content data. It is another object to make digital content more difficult to replicate or to render by unauthorized agents. It is another object of the invention to provide a secure and forward compatible method of digital content distribution.

5   To this end, the invention provides a method of providing or receiving content information, preferably via a network, as one or more software objects. A procedure is encapsulated in the object for access of the information on a runtime environment. That is, the procedure for accessing the information by the end-user is packaged together with the content information for delivery to the end-user.

10   As known, a software object comprises content data plus a method or procedure to process the data. In object-oriented programming, a method is a programmed procedure that is defined as part of a class and that is included in any object of that class. An object can have more than one method. A method in an object can only have access to the data known to that object, which ensures data integrity among the set of objects in an application.

15   Open distributed software architectures such as HAVi, Java/JINI, Home API based on COM or DCOM technology, CORBA and others allow transfer of software objects over a network (Internet, home network, local area network, wide area network, etc.). For more information on software representation, HAVi, the use of COM technology and OLE Automation objects, the following patent documents are incorporated herein by reference:

20   U.S. Serial No. 08/731,624 (Attorney docket PHA 23,169) filed 10/15/96 for Paul Chambers and Saurabh Srivastava for "TASK-DRIVEN DISTRIBUTED MULTIMEDIA CONSUMER SYSTEM"; U.S. Serial No. 09/146,020 (Attorney Docket PHA 23,492), filed 9/2/98 for Yevgeniy Shteyn, for "LOW DATA-RATE NETWORK REPRESENTED ON HIGH DATA-RATE HAVi-NETWORK"; U.S. Serial No. 09/165,683 (Attorney Docket PHA

25   23,483), filed 10/2/98 for Yevgeniy Shteyn for "CALLS IDENTIFY SCENARIO FOR CONTROL OF SOFTWARE OBJECTS VIA PROPERTY ROUTES"; U.S. Serial No. 09/165,682 (Attorney Docket PHA 23,484), filed 10/2/98 for Yevgeniy Shteyn for "CONTROL PROPERTY IS MAPPED ONTO MODALLY COMPATIBLE GUI ELEMENT"; and U.S. Serial No. 09/107,525 (Attorney Docket PHA 23,438), filed 6/30/98

30   for Yevgenyi Shteyn and Gregory Gewickey for "DYNAMIC DE-REGISTERING OF DEVICES IN SYSTEM WITH MULTIPLE COMMUNICATION PROTOCOLS".

When transferred to a machine such software objects can interact with its system software. For example, a HAVi DCM can be uploaded to an FAV and executed in a Java run-time environment. The DCM can be programmed to implement API's for the

rendering, storing, recording or other processing of graphics, audio, video, text, image, etc. In this case, the software object works directly with the rendering machine, without intermediate decryption or modification. All operations are controlled by the object itself and hidden from hostile applications. When content needs to be stored, an object-controlled storage API is

5   invoked (for example: Serialization, see further below). Since the object controls the storage (recording) process, the host system has to recreate the object in order to access stored content and therefore its content is protected. The object can affect the quality of recording, the number of times the content can be recorded or rendered and other interactions with the host system. The software object also can be made time-sensitive. For example, a movie object can

10   be leased, i.e., its rendering API can only be used for a certain amount of time – movie playing time, a day, etc…A controlling object does not have to contain all the content. When the content is very large, the object could just keep the reference(s) to content location and does the retrieval as needed. Where necessary or desirable, a chain of objects could control operations. For example, different movie scenes are controlled by different objects. An object

15   can be set up to provide free (trial) access to certain parts of the content and require payments for other. Conditional interaction with system resources, based on user preferences, can be accomplished. Advertisements and promotions can be inserted as separate objects, depending on the level of user access (e.g., via subscription).

An object can wrap up binary code, provided the execution environment of the

20   target client is known or can be detected after object interaction with the device. For example, a better rendering algorithm or a binary upgrade can be delivered to a TriMedia machine running Java VM.

The proposed method of data content delivery to an end-user through objects is also highly suitable for other usages than copyright protection. For example, email,

25   teleconferencing (via telephone) or videoconferencing can be based on the exchange of software objects for security reasons. Similarly, electronic monetary transactions are made secure through the communication of software objects. Electronic mail can be implemented as software objects.

Transport of an object over a network, e.g., from the server to the client, is

30   accomplished through, e.g., object serialization. Object serialization in Java supports the encoding of objects, and the objects reachable from them, into a stream of bytes, and it supports the complementary reconstruction of the object graph from the stream. Serialization is used for lightweight persistence and for communication via sockets or Remote Method Invocation (RMI). The default encoding of objects protects private and transient data, and

supports the evolution of the classes. A separate digital data stream that can be pirated and replicated by hostile parties does not exist in the traditional sense. All what is being sent is the string of bits and it is not possible to the average person to find a clue about its meaning, since procedures and content information are not recognizable from the stream. A class may

5 implement its own external encoding to increase security and is then solely responsible for the external format (source: http:// www.javasoft.com/ products /jdk/1.1/ docs/guide/ serialization/ index.html). Accordingly, a decryption procedure can be encapsulated in the object, together with encrypted content data. The decryption procedure is then invoked at the receiving end, possibly complemented with an also encapsulated verification procedure verifying the

10 receiver's ID, decryption key, or serial number of the rendering device, etc.

For the sake of clarity, the downloading of an Applet via the Internet is known. User-access to the Applet requires a browser application residing on the receiving client. The Applet gets executed in the browser application. The browser or plug-in is not part of the Applet. In the invention, the software object gets activated directly in the runtime environment

15 of the receiver (e.g., HAVi runtime Home API runtime [=Windows OS]), Jini runtime) using its own procedures and does not require a browser or any client application in the runtime.

Various advantageous aspects of the invention are as listed in the appended claims and any viable combination thereof. For example, a video conferencing system can provide conventional video streaming and object-encapsulated secure audio and/or graphical

20 data.

The invention is explained in further detail and by way of example with reference to the accompanying drawing, wherein:

25 Figs.1-5 are block diagrams of examples of a system wherein the method of the invention is implemented.

Throughout the drawing same reference numerals indicate similar or corresponding features.

30

Fig.1 is a block diagram of a HAVi system 100 for implementing the method of the invention. System 100 comprises a storage medium 102 and an FAV (Full AV node) 104 that comprises a digital TV in this example. FAV 104 has a Java Virtual Machine 106. Storage medium 102 comprises, for example, a buffer that stores a software object 108 received

electronically from outside system 100, e.g., via the Internet after de-serialization, or a DVD

or another physical data-carrier that contains the software object, here a HAVi DCM 108 with

content data 110 and play-out procedures, represented by rendering API 112. Software object

108 is uploaded to the rendering process on FAV 104 where rendering API 104 directly

5   interacts with Java VM 106. Conditional access to the content can be based on security or on

user/system capabilities. In a security-based model, the object allows invocation for play-out

or storage through API's dedicated to verification. The run time would have to provide an

authorization or an identifier that can be verified by the object. For example, parental control

can be implemented by a group of objects representing different parts of a movie content. As

10   another example, an electronic book can encapsulate graphical objects with access restriction

requirements.

Fig.2 is a block diagram of a Jini system 200 which uses a JavaSpaces

architecture. JavaSpaces is an architecture for creating a distributed operating system and

creating repositories for Java-based objects, as well as Java VM's and Java's Remote Method

15   Invocation (RMI) functionality. JavaSpaces implementations provide a mechanism for storing

a group of related objects and retrieving them based on value-matching look-up for specified

fields. This allows a JavaSpaces server to be used for storing and retrieving objects on a

remote system. The JavaSpaces API uses a specific package to provide basic atomic

transactions that group multiple operations across multiple JavaSpaces implementations into a

20   bundle that acts as a single atomic operation. The JavaSpaces architecture supports a

transaction mechanism allows multi-operation and/or multi-space updates to complete

atomically. The term "atomicity" means that all operations grouped under a transaction occur

or none of them does. Jini is a technology, also from Sun Microsystems, for networking of

devices. Jini is a Java-based software technology that assists in networking PC's and

25   peripherals. When plugged into a network, a Jini-enabled device will broadcast its presence.

Network clients that are ready to use that device can request the necessary software from the

device itself, bypassing a server or a network administrator. This architecture builds on top of

an existing network.

System 200 comprises JavaSpaces servers 202 and 204, and a client 206

30   connected through a network 208. JavaSpaces server 202 or 204 provides a distributed

persistence and object exchange mechanism for objects written in the Java programming

language. Objects are written in entries that provide a typed grouping of relevant fields.

Clients can perform simple operations on a JavaSpaces server to write new new entries, to

look-up existing entries, and remove entries from the space. Using these tools a user can write

systems that use flow of data to implement distributed algorithms and let the JavaSpaces
system implement persistence for the user. Upon an appropriate request, server 202 transfers
an object (through Serialization) to client 206. The transfer is a lease as specified by Java's
Distributed Leasing Specification. Rather than granting services or resources until that grant
5   has been explicitly canceled by whom the grant was made, a leased resource or service grant is
time-based. When the time for the lease has expired the service ends or the resource is freed.
This notion of "lease" is especially useful in distributed systems, where different parts of a
cooperating group are unable to communicate, e.g., because of a failure of a member of the
group or of the connection. According to the diagram of Fig.2, server 202 leases a first object
10   to server 204 for temporary storage and a second object to client 206 for temporary storage or
rendering. This configuration is relevant to, for example, a video- (or audio-) on-demand
service provider, a multi-player video game provider, etc. A similar client-server architecture
can be built using COM/DCOM technology of Microsoft, with a COM client on a Windows
operating system.

15   Fig.3 is a diagram of a system 300 with a COM/DCOM server 302 and a COM
client 304. Server 302 provides object access and object storage. Client 304 can access server
302 using COM or DCOM mechanisms. For a remote location, DCOM is be used. The access
mechanism is transparent to client 304. A COM object is transferred into the process on client
304 and an appropriate API is invoked at client 304 in order to produce desirable content. The
20   object has full access to system resources and API's such as DirectShow components from
Microsoft.

            Fig.4 is a diagram of another configuration of a system 400 in the invention.
System 400 comprises a server 402 for providing objects and serving as an object repository.
System 400 further comprises clients 404 and 406. System 400 is used for, e.g., multi-client
25   interaction for exchange of content such as in videoconferencing. Content objects are placed in
server/repository 402 by client 404 and can be retrieved from repository 402 by client 406.

            Fig.5 is a diagram of a system 500 that has a similar configuration as system
400. System 500 comprises an object server 502 that serves clients 504, 506, 508 and 510.
Server 502 functions as an object repository to enable multi-client object interaction. Content
30   objects are placed in server 502 by any of clients 504-510 and can be retrieved by any of the
other clients 504-510. System 500 comprises a relay or replication server 512 to serve clients
514 and 516. A content object can be packaged or can expose interfaces based on preferences
or access privileges of clients 504-510 and 514-516.

For more information, reference is made to the relevant Specifications of Java, JavaSpaces and Jini Architecture, all of Sun Microsystems, and of the Component Object Model Specification of Microsoft, available in the public domain and herein incorporated by reference.

CLAIMS:

1.   A method of supplying to an end-user content information (110) in a software object (108) that has an encapsulated procedure (112) for end-user access of the information in a runtime environment (104).

2.   The method of claim 1, wherein the procedure enables conditional access to the content information.

3.   The method of claim 1, comprising serializing the software object for providing the software object electronically over a network (208).

4.   The method of claim 1, wherein the object contains a procedure for enabling play-out of the content information at a play-out device (104).

5.   The method of claim 1, wherein the object contains a procedure for enabling storing the content information in a memory (204; 206).

6.   The method of claim 1, wherein the providing is comprised in an electronic mail service.

7.   The method of claim 1, wherein the providing is comprised in a teleconference service.

8.   The method of claim 1, wherein the providing is comprised in a videoconference service.

9.   The method of claim 1, wherein the content information comprises audio information.

10.   The method of claim 1, wherein the content information comprises video information.

11.   ·The method of claim 1, wherein the content comprises a movie.

5

12.   A method of receiving content information as one or more software objects that contain a procedure for enabling an end-user to access the content information in a runtime environment (104).

10   13.   The method of claim 12, comprising enabling conditional access to the content information through a conditional access procedure in the object.

14.   The method of claim 12, wherein the content information comprises audio information.

15

15.   The method of claim 12, wherein the content information comprises video information.

16.   The method of claim 12, wherein the content information comprises a movie.

20

17.   The method of claim 12, wherein the content information comprises alphanumeric information.

18.   A method of preparing content information for an end-user, the method
25   comprising packaging the content information (110) in a software object that has encapsulated a procedure (112) for enabling the end-user to access the information in a runtime environment (104).

19.   The method of claim 18, wherein the content information comprises a movie.

30

20.   The method of claim 18, wherein the content information comprises alphanumeric information.
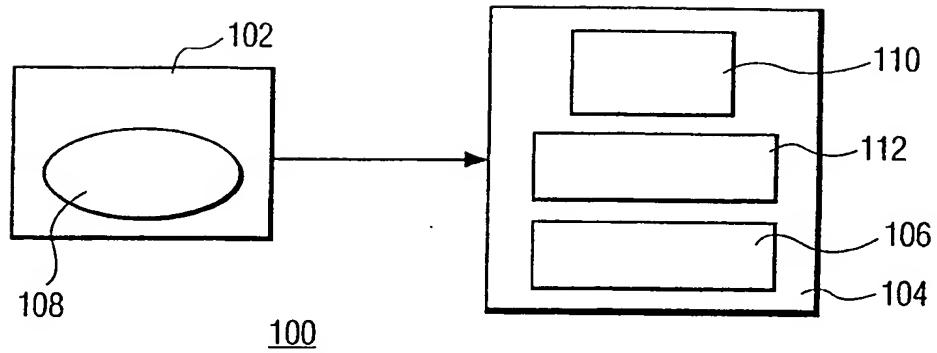
FIG. 1
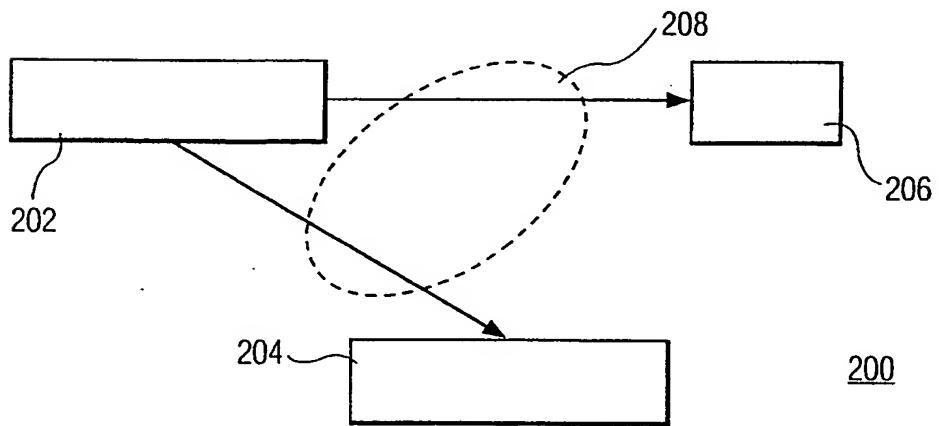
FIG. 2
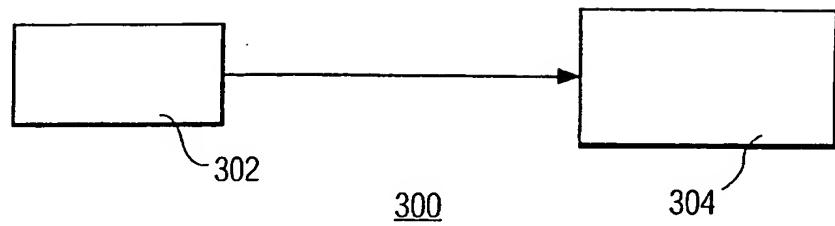
FIG. 3
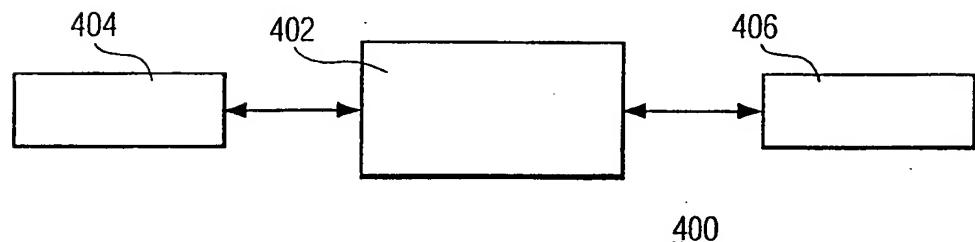
FIG. 4
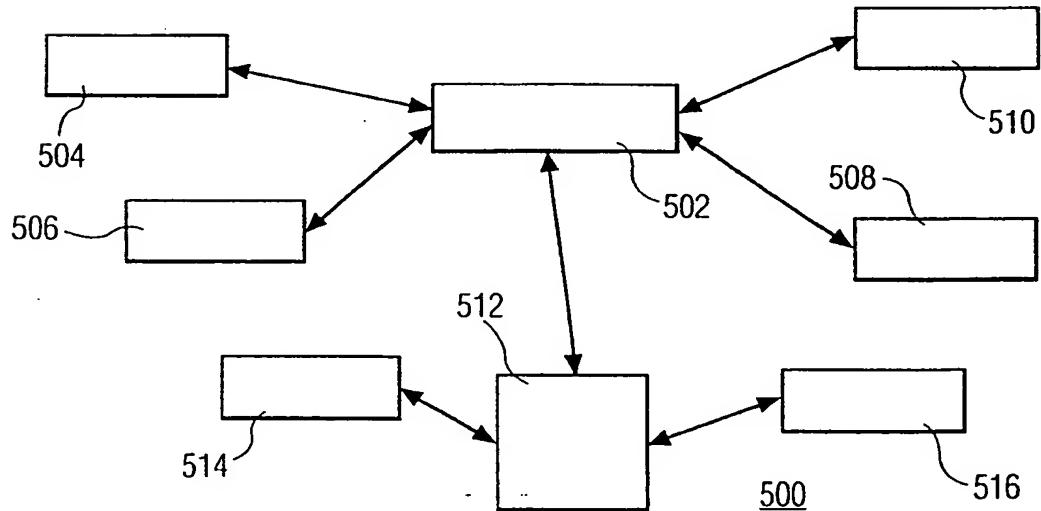
FIG. 5

INTERNATIONAL SEARCH REPORT

A. CLASSIFICATION OF SUBJECT MATTER

IPC 7   G06F1/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 7   G06F   H04N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category°	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	JAVA SECURITY, 'Online! XP002092325 Retrieved from the Internet: <URL:http://SWISSNET.AI.MIT.EDU/{JBANK/JAVAPAPER/JAVAPAPER.HTML> 'retrieved on 1999-02-04! page 1, line 19 – line 32 ---	1,3-6, 12,17,. 18,20
A	US 5 420 866 A (WASILEWSKI ANTHONY J) 30 May 1995 (1995-05-30) figures 2,7 column 7, line 64 –column 8, line 30 column 15, line 18 –column 16, line 64 -----	1-5,9-20

☐ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
20 March 2000	28/03/2000

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31–70) 340–2040, Tx. 31 651 epo nl, Fax: (+31–70) 340–3016	Weiss, P

Form PCT/ISA/210 (second sheet) (July 1992)

1

Patent document cited in search report		Publication date	Patent family member(s)		Publication date
US 5420866	A	30-05-1995	AU	687844 B	05-03-1998
			AU	7220994 A	17-10-1995
			CA	2186368 A,C	05-10-1995
			JP	2940639 B	25-08-1999
			JP	9511369 T	11-11-1997
			WO	9526597 A	05-10-1995